

## CLAIMS

WE CLAIM:

1. A method of recovering an application using a timeline-based computing environment, the method comprising the steps of:  
  
periodically saving data items of the application for recovery, wherein the saved data items allow the recovery of the application at a point in time when the items were saved;  
  
searching a time-based archive of the saved data items;  
  
designating a point in time from which to recover the application; and  
  
recovering the application using the data items saved at the designated point in time.
2. The method according to claim 1 further comprising the step of saving contextual information relevant to the point in time when the data items are saved.
3. The method according to claim 1 further comprising the step of generating a fork of the recovered application at the designated point for usage.
4. The method according to claim 3 wherein the generated fork is an alternative timeline reflecting new usage of the recovered application.
5. The method according to claim 1 wherein the saving step further comprises the steps of:

determining whether the application is of a type that can save a working file without quitting the application;

observing the application for any working file that is being saved by the application when the application is of the type that can save the working file without quitting the application; and

capturing the saved working file.

6. The method according to claim 1 wherein the saving step further comprises the steps of:

determining whether the application is of a type that saves a recovery file without quitting the application and requires the application to quit in order to save a working file;

observing the application for any recovery file that is being saved by the application when the application is of the type that saves the recovery file without quitting the application and requires the application to quit in order to save the working file;

capturing the saved recovery file; and

capturing a mapping between the captured recovery file and the working file.

7. The method according to claim 1 wherein the saving step further comprises the steps of:

determining whether the application is of a type that must quit before a working file can be saved;

creating a clone of the application when the application is of the type that must quit before the working file can be saved;

sending a quit command to the clone of the application to initiate a save file command; and

capturing any saved files responsive to the save file command.

8. The method according to claim 1 wherein the saving step further comprises the steps of:

determining whether there is sufficient space to capture a machine state;

running a virtual machine when there is sufficient space to capture the machine state;

executing the application in the virtual machine; and

capturing the state of the virtual machine at the point in time.

9. The method according to claim 1 wherein the searching step comprises the steps of:

indexing metadata for each saved data item; and

generating a list of data items according to a match between the indexed metadata and a user selected variable.

10. The method according to claim 9 wherein the indexing step further comprises the step of generating the metadata according to a user action or an automatic algorithm that catalogues the saved data items.

11. The method according to claim 1 wherein the searching step comprises the steps of:

scrolling on a timeline indicating different points in time; and

changing a graphical representation of the application according to the scrolling step.

12. The method according to claim 11 wherein the scrolling provides an indication of degree, volume, or type of the saved data items.

13. The method according to claim 1 wherein the application is a communication client having a plurality of messages, the method further comprising the steps of:

saving an index data to indicate whether a message in the communication client is spam; and

generating a list of messages according to a match between the plurality of messages that are not indicated as spam and a user selected search variable.

14. The method according to claim 1 wherein the application is a communication client having a plurality of messages, the method further comprising the steps of:

marking an index data to indicate whether a message in the communication client is spam;

purging all messages marked as spam prior to the step of periodically saving data items of the application for recovery.

15. The method according to claim 1 further comprising the steps of:  
copying data from the application recovered at the designated point in time;  
returning to a current point in time; and  
pasting the copied data in the application in the current point in time.
16. A computer-readable medium having thereon computer-executable instructions for recovering an application using a timeline-based computing environment, the instructions for performing steps comprising:  
periodically saving data items of the application for recovery, wherein the saved data items allow the recovery of the application at a point in time when the items were saved;  
searching a time-based archive of the saved data items;  
designating a point in time from which to recover the application; and  
recovering the application using the data items saved at the designated point in time.
17. The medium according to claim 16 further comprising computer-executable instructions for saving contextual information relevant to the point in time when the data items are saved.

18. The medium according to claim 16 further comprising computer-executable instructions for generating a fork of the recovered application at the designated point for usage.

19. The medium according to claim 18 wherein the generated fork is an alternative timeline reflecting new usage of the recovered application.

20. The medium according to claim 16 wherein the computer-executable instructions for periodically saving data items further comprise computer-executable instructions for:  
determining whether the application is of a type that can save a working file without quitting the application;

observing the application for any working file that is being saved by the application when the application is of the type that can save the working file without quitting the application; and

capturing the saved working file.

21. The medium according to claim 16 wherein the computer-executable instructions for periodically saving data items further comprise computer-executable instructions for:

determining whether the application is of a type that saves a recovery file without quitting the application and requires the application to quit in order to save a working file;

observing the application for any recovery file that is being saved by the application when the application is of the type that saves the recovery file without

quitting the application and requires the application to quit in order to save the working file;

capturing the saved recovery file; and

capturing a mapping between the captured recovery file and the working file.

22. The medium according to claim 16 wherein the computer-executable instructions for periodically saving data items further comprise computer-executable instructions for:

determining whether the application is of a type that must quit before a working file can be saved;

creating a clone of the application when the application is of the type that must quit before the working file can be saved;

sending a quit command to the clone of the application to initiate a save file command; and

capturing any saved files responsive to the save file command.

23. The medium according to claim 16 wherein the computer-executable instructions for periodically saving data items further comprise computer-executable instructions for:

determining whether there is sufficient space to capture a machine state;

running a virtual machine when there is sufficient space to capture the machine state;

executing the application in the virtual machine; and

capturing the state of the virtual machine at the point in time.

24. The medium according to claim 16 wherein the computer-executable instructions for searching a time-based archive further comprising executable instructions for:

indexing metadata for each saved data item; and

generating a list of data items according to a match between the indexed metadata and a user selected variable.

25. The medium according to claim 24 wherein the computer-executable instructions for indexing metadata comprising executable instructions for generating the metadata according to a user action or an automatic algorithm that catalogues the saved data items.

26. The medium according to claim 16 wherein the computer-executable instructions for searching a time-based archive further comprising executable instructions for:

scrolling on a timeline indicating different points in time; and

changing a graphical representation of the application according the scrolling step.

27. The medium according to claim 26 wherein the computer-executable instructions for scrolling on a timeline provides an indication of degree, volume, or type of the saved data items.

28. The medium according to claim 16 wherein the application is a communication client having a plurality of messages, the medium further comprises computer-executable instructions for:



saving an index data to indicate whether a message in the communication client is spam; and

generating a list of messages according to a match between the plurality of messages that are not indicated as spam and a user selected search variable.

29. The medium according to claim 16 wherein the application is a communication client having a plurality of messages, the medium further comprises computer-executable instructions for:

marking an index data to indicate whether a message in the communication client is spam;

purging all messages marked as spam prior to the step of periodically saving data items of the application for recovery.

30. The medium according to claim 16 further comprises computer-executable instructions for:

copying data from the application recovered at the designated point in time;

returning to a current point in time; and

pasting the copied data in the application in the current point in time.

31. In a computer system having a graphical user interface including a display and a user interface selection device, a method of providing and selecting from a menu on the display for recovering an application using a timeline-based computing environment, comprising the steps of:

- copying data from the application recovered at a designated point in time;
- moving the application to a current point in time; and
- pasting the copied data to the application in the current point in time.
32. The computer system according to claim 31 further comprising the steps of:
- displaying a window graphical representation of the application with a pointer indicating different points in time; and
- recovering the application at a designated point in time.
33. The computer system according to claim 32 wherein the window graphical representation of the application represents visual alternative application states via forking on a timeline.